MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

AD-A244 545

...ENTATION PAGE

Form Approved OMB No. 0704-0188

...mated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, ...d reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this ...s burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson ...e Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

...PORT DATE: November 1991

3. REPORT TYPE AND DATES COVERED: Reprint (July - December 1991)

4. TITLE AND SUBTITLE
Design of Pipelined Lattice IIR Digital Filters

5. FUNDING NUMBERS
DAAL03-90-G-0063

6. AUTHOR(S)
J. G. Chung and K. K.Parhi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Minnesota
Department of Electrical Engineering
200 Union Street SE
Minneapolis, MN 55455

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARO 27076.20-EL

11. SUPPLEMENTARY NOTES
The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Lattice structures exhibit good finite word length properties. In some real-time applications, high speed is very crucial to the system performance along with finite word length effects. Although lattice filters can be pipelined by cut-set localization procedure, no sample rate increase can be achieved by that method.

In this paper, a pipelining method in lattice digital filters is introduced. This pipelining method is based upon constrained IIR digital filter design method by which pipelined direct-form filters are designed. These direct-form filters are transformed to pipelined lattice digital filters. It is shown that the roundoff error and the number of multiply/add operations of the resulting pipelined lattice filters are smaller than those of the pipelined lattice filters obtained by applying look-ahead on direct-form nonpipelined digital filters.

92-00802

14. SUBJECT TERMS
IIR Digital Filters, Lattice Filters, Pipelining, High-Speed, Constrained Filter Design, Finite Word Effects

15. NUMBER OF PAGES
5

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Proceedings of the Twenty - Fifth Asilomar Conference on Signals, Systems and Computers
November 4 - 6, 1991, Pacific Grove (CA)

Design of Pipelined Lattice IIR Digital Filters

Jin-Gyun Chung, and Keshab K. Parhi

Department of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455, USA

Abstract

Lattice structures exhibit good finite word length properties. In some real-time applications, high speed is very crucial to the system performance along with finite word length effects. Although lattice filters can be pipelined by cut-set localization procedure, no sample rate increase can be achieved by that method.

In this paper, a pipelining method in lattice digital filters is introduced. This pipelining method is based upon constrained IIR digital filter design method by which pipelined direct-form filters are designed. These direct-form filters are transformed to pipelined lattice digital filters. It is shown that the roundoff error and the number of multiply/add operations of the resulting pipelined lattice filters are smaller than those of the pipelined lattice filters obtained by applying look-ahead on direct-form nonpipelined digital filters.

1. Introduction

Since lattice structures exhibit very good finite word length properties, these structures are preferable for fixed-point finite word-length implementation. Different lattice digital filter structures have been proposed in [1-4].

Achieving high speed in direct-form and and lattice digital filters is difficult because of the feedback loop. Although these filters can be pipelined by cut-set localization procedure proposed in [5], it should be noted that no sample rate increase can be achieved by that method. Cutset transformation can increase the clock speed, but cannot increase the sample rate, since multiple clock cycles are needed to process one sample.

Implementation of high-speed, pipelined, stable, recursive digital filters was considered impossible before. Only recently, scattered look-ahead techniques were proposed to convert a nonpipelined filter to a pipelinable form [6]. The scattered look-ahead technique adds poles at equal angular and radial spacing for each pole of the nonpipelined filter, and these poles are canceled by zeros. This transformation converts the denominator of the transfer function to a pipelined form. This technique, for the first time, made it feasible to implement high speed recursive digital filters using pipelining.

One drawback of the scattered look-ahead technique is the introduction of the canceling zeros. These zeros increase the number of multiplication operations needed to implement the digital filter. Therefore, as suggested in [6], it is of interest to design pipelinable digital filter transfer functions directly from the filter spectrum, as opposed to designing a nonpipelined filter and applying scattered look-ahead on this filter.

To this end, we address the design of pipelined direct-form recursive digital filters from the filter spectrum. By constraining the location of poles at equal angular and radial spacing, we can design a transfer function whose denominator is pipelinable. Fortunately, this is possible by using a variation of the filter design procedure used for decimation filter design [7-10]. The pipelinable filter transfer function can be mapped to many lattice digital filter structures. Pipelining the orthogonal structure in [3] leads to no increase in speed. Therefore, the orthogonal lattice filter structure is not suitable for pipelining. The minimum-noise lattice structure in [4] requires more number of muiltiply-add operations than [1], but leads to less roundoff error. In this paper, we propose the use of pipelining for the basic, normalized, and the minimum-noise lattice structure.

This paper is organized as follows. The Schur algorithm [11] which forms the basis for design of lattice digital filters is reviewed in section 2. Section 3 describes the theory of lattice filter pipelining. Section 4 presents the constrained filter design procedure for design of pipelined direct-form recursive digital filters, which are then reduced to lattice structures. Section 5 presents design examples for several lattice filters.

2. Schur Algorithm

One of the important properties of the Schur algorithm is that *all polynomials expanded from any given polynomial by the Schur algorithm are orthogonal to each other*. Schur polynomial is a polynomial which

This research was spported by the army research office by contract number DAAL03-90-G-0063.

does not have zeros on or outside the unit circle. Therefore, the denominator of a stable IIR filter is a Schur polynomial. Let a polynomial with real coefficients be defined by

$$D(z) = \sum_{i=0}^{N} d_i z^i.$$

Then, initialize *N*th-order Schur polynomial $\Phi_N(z)$ as $\Phi_N(z) = D(z)$,

$$\text{where } \Phi_N(z) = \sum_{i=0}^{N} \phi_i z^i.$$

From $\Phi_N(z)$, form the polynomial $\Phi_{N-1}(z)$ as follows:

$$\Phi_{N-1}(z) = \frac{z^{-1}[\phi_n \Phi_N(z) - \phi_0 \Phi_N^*(z)]}{\sqrt{\phi_N^2 - \phi_0^2}},$$

where $\Phi_N^*(z)$ is the reverse polynomial of $\Phi_N(z)$ and is defined by $z^N \Phi_N(z^{-1})$. The degree of $\Phi_{N-1}(z)$ is 1 less than $\Phi_N(z)$.

In order to implement the algorithm, each of the polynomials $\Phi_{N-i}(z)$ is generated from $\Phi_{N-i+1}(z)$, for $i=1,2, \ldots ,N$, in the same manner as $\Phi_{N-1}(z)$ is generated from $\Phi_N(z)$. It may be noted that the coefficients of increasing powers of z in $\Phi_{N-1}(z)$ are the N determinants of 2 x 2 submatrices formed by the first column and each succeeding column in the following matrix.

$$\frac{1}{C}\begin{bmatrix} \phi_N & \phi_{N-1} & \phi_{N-2} & \cdots & \phi_1 & \phi_0 \\ \phi_0 & \phi_1 & \phi_2 & \cdots & \phi_{N-1} & \phi_N \end{bmatrix},$$

$$\text{where } C = \sqrt{\phi_N^2 - \phi_0^2}.$$

For each 2 x 2 submatrix, if a column is composed of all zero elements, the determinant of the corresponding submatrix is zero. Using this property of determinant, we can observe an useful fact for pipelining:
If $\Phi_N(z)$ is a Schur polynomial of order N, and has j-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then $\Phi_{N-i}(z)$ of order N-i also has j-consecutive zero coefficients between each two nonzero coefficients of nearest degree for i = 1 to N-j-1 and has only one nonzero coefficient for i = N-j to N.

3. Pipelining of Recursive Digital Lattice Filters

In this section, we study pipelinability of basic, normalized and minimum-noise lattice filters.

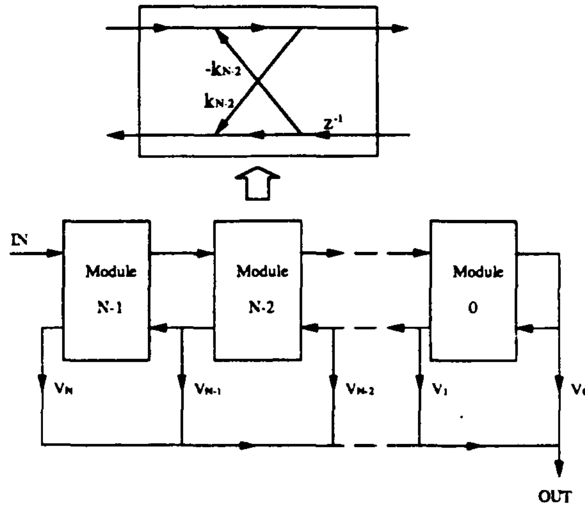

Fig.1 The structure of Basic lattice filter

3.1 Pipelining of Basic Lattice Filter

Fig. 1 shows the structure of the basic lattice filter which is characterized by *N* k-parameters and *N+1* tap gain parameters. The multiply-add operations in the feed-forward section can be pipelined at any desired level by placing latches at appropriate feed-forward cutset locations. As can be seen from Fig. 1, the feedback loop in a module is removed if the k-parameter of the module is zero. Therefore, by removing *(M - 1)*-consecutive k-parameters, the basic lattice filter can be pipelined by *M*-stages, leading to *M* times increase in speed as compared with the original basic lattice filter.

Let the *N*th-order denominator of a digital filter transfer function be denoted by

$$D_m(z) = \sum_{i=0}^{m} d_{m,i} z^{-i},$$

where $m = N$, and the leading coefficient, $d_{m,0}$, is assumed to be one. Then, k-parameters are recursively obtained by starting from $D_N(z)$ using:

1) $k_{m-1} = d_{m,m}$

2) $D_{m-1}(z) = \dfrac{D_m(z) - k_{m-1} D_m^*(z)}{1-k_{m-1}^2}$

for $m = N, N-1, \ldots , 1$.

As can be seen from above steps, $D_m(z)$'s for $m = N, N-1, \ldots, 0$ are Schur polynomials with a different normalization constant. Therefore, if $D_N(z)$ has *(M - 1)*-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then $D_m(z)$'s for $m = N, N-1, \ldots, M$, also have *(M - 1)*-consecutive zero coefficients between each two nonzero coefficients of nearest degree

and $D_m(z)$'s for $m = M-1, M-2, ..., 1$, have only one nonzero coefficients. Then, since k-parameters are the coefficients of z^{-m} of each $D_m(z)$ for $m=N, N-1, ..., 1$, we can obtain $(M-1)$-consecutive zero k-parameters, which leads to M-stage pipelining. In general, an M-stage pipelined filter contains M-delays in every loop. Since the number of multiply-add operations in every loop remains constant, the M-stage pipelined filter can be clocked at M-times faster rate, as compared with the nonpipelined filter.

3.2 Pipelining of Normalized Lattice Filter

The structure of normalized lattice filter is shown in Fig. 2, where the feedback section is again described by k-parameters only. The k-parameters are calculated by the same procedure as in section 3.1.

As can be seen from Fig. 2, there are four multipliers within a module m: k_m, $-k_m$, and two $\sqrt{1-k_m^2}$'s. If k_m is zero, then $\sqrt{1-k_m^2}$ becomes one. Therefore, module m needs no multipliers and the feedback loop of the module is removed, which means the normalized lattice filter can be pipelined.

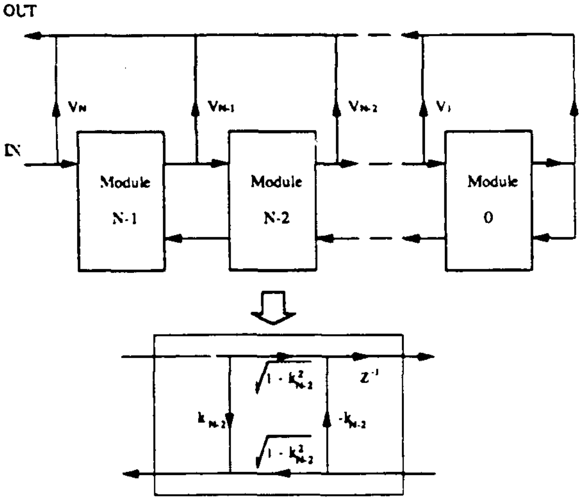

Fig. 2 The structure of Normalized lattice filter

3.3 Pipelining of Minimum-Noise Lattice Filter

The structure of the minimum-noise lattice filter [4] is shown in Fig. 3. Feedback section is described by k-parameters only and the k-parameters are calculated using Schur polynomials as follows:

$$k_m = \frac{\Phi_m(0)}{\Phi_m^*(0)},$$

where $\Phi_m(z)$ is mth-order Schur polynomial obtained from the denominator. If k_m of the module m is zero, then the feedback loop of the module is removed since $\sqrt{1-k_m^2}$ becomes one. For k_m to be zero, $\Phi_m(0)$, or the constant term of $\Phi_m(z)$, should be zero. As was discussed in section 2, if the denominator has $(M-1)$-consecutive zero coefficients between each two nonzero coefficients of nearest degree, then the minimum-noise filter can be pipelined.

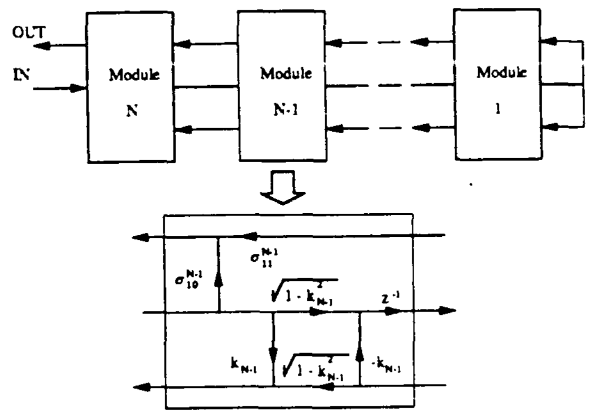

Fig. 3 The structure of Minimum-noise lattice filter

4. Design of Pipelined IIR Lattice Filter Using Constrained Filter Design

It is apparent from previous sections that the basic, normalized, and minimum-noise lattice filters can be pipelined by M-stages if the denominator of a transfer function has $(M-1)$-consecutive zero coefficients between each two nonzero coefficients of nearest degree. The pipelinable transfer functions can be obtained by applying the scattered look-ahead method to the nonpipelined filter transfer functions. However, by using constrained filter design method, we can design inherently pipelinable transfer functions directly from the filter spectrum specifications.

Our procedure is a variation of the procedures in [7] and [9]. The design method in [7] first expresses the magnitude and group delay responses of the filter as functions of the radii and angles of the poles and zeros. Then we obtain formulae for the partial derivatives of the magnitude and group delay with respect to the radius and the angle of a pole and a zero. These derivatives are used in the Fletcher-Powell algorithm to minimize the approximation error [8]. In this algorithm, the minimum value is obtained by only using the first derivative information. To obtain an inherently M-stage pipelinable filter, we recompute the partial derivatives for a denominator in powers of z^M rather than z [9] and substitute these equations into the program in [10].

Then, the resulted denominator is in terms of z^{-M}.

The modified filter design procedure is summarized as follows:

1) For the given filter specifications (pass-band, stop-band, pass-band & stop-band ripples, and M-stage pipelining), we start with 1-complex pole pair. Then, the order of the denominator is $2M$ and the number of zeros are restricted so that the order of the numerator is less than or equal to $2M$. We usually start with M unit-circle zero pairs.

2) If the filter specifications are not satisfied by 1-complex pole pair, we increase the number of poles and zeros and/or adjust initial positions of poles and zeros and weighting factors for the pass-band and stop-band. The filter is redesigned. This procedure is repeated until the filter specifications are satisfied.

3) The pipelined direct-form filter transfer function is then used to obtain the pipelined lattice filters by using the sybthesis procedures of section 3.

5. Design Example of Pipelined Lattice Filters

Consider the lowpass filter with 1)pass-band: $0 \div 0.2\pi$, ripple: 1 dB, and 2)stop-band: $0.3\pi - \pi$, ripple: 15 dB. In [12], this example was designed by using the 4th order Chebyshev filter:

$N(z) = 0.0018 + 0.0073z^{-1} + 0.0011z^{-2} + 0.0073z^{-3} + 0.0018z^{-4}$

$D(z) = 1 - 3.0544z^{-1} + 3.829z^{-2} - 2.2925z^{-3} + 0.05z^{-4}$,

where N(z) and D(z) denote numerator and denominator of the transfer function, respectively.

Using the constrained filter design procedure, we obtained the transfer functions for different number of pipelining stages. These filters are listed below.

1) Nonpipelined

$N(z) = 0.0307 + 0.0054z^{-1} + 0.0203z^{-2} + 0.0054z^{-3} + 0.0307z^{-4}$

$D(z) = 1 - 2.5754z^{-1} + 2.868z^{-2} - 1.5335z^{-3} + 0.333z^{-4}$.

2) 2-stage pipelined (M = 2)

$N(z) = 0.0532 + 0.112z^{-1} + 0.1159z^{-2} + 0.0941z^{-3} + 0.1159z^{-4} + 0.112z^{-5} + 0.0532z^{-6}$

$D(z) = 1 - 1.02z^{-2} + 1.065z^{-4} - 0.509z^{-6} + 0.128z^{-8}$.

3) 4-stage pipelined (M = 4)

$N(z) = 0.0754 + 0.132z^{-1} + 0.2018z^{-2} + 0.2411z^{-3} + 0.2815z^{-4} + 0.2411z^{-5} + 0.2018z^{-6} + 0.132z^{-7} + 0.10754z^{-8}$.

$D(z) = 1 + 0.5626z^{-4} + 0.0791z^{-8}$.

4) 8-stage pipelined (M = 8)

$N(z) = 0.0974 + 0.0982z^{-1} + 0.0814z^{-2} + 0.0347z^{-3} + 0.0047z^{-4} - 0.0275z^{-5} + 0.029z^{-6} + 0.0761z^{-7} + 0.065z^{-8} + 0.012z^{-9} - 0.0434z^{-10} - 0.058z^{-11} - 0.055z^{-12} - 0.0034z^{-13} + 0.0082z^{-14} + 0.0098z^{-15} + 0.0098z^{-16}$

$D(z) = 1 - 0.2013z^{-8} + 0.101z^{-16}$.

Fig. 4 and Fig.5 show the magnitude response and the implementation of the 4-stage pipelined normalized lattice filter obtained from the above transfer function.

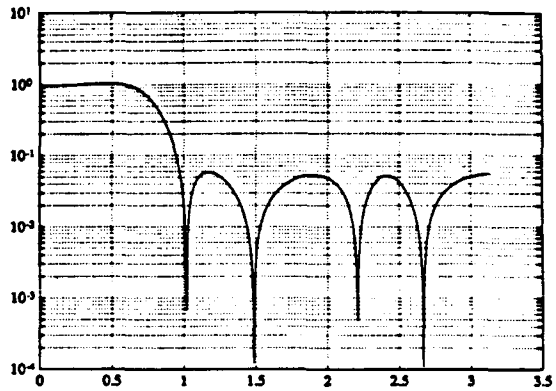

Fig. 4 Magnitude response of 4-stage pipelined filter by Constrained design

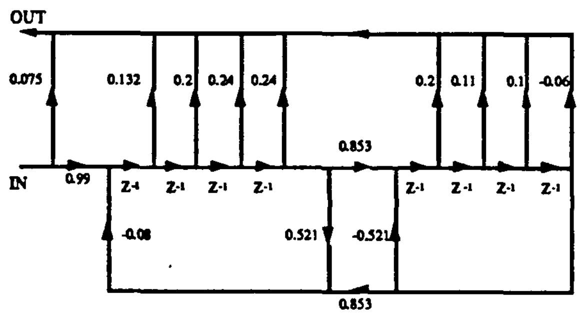

Fig. 5 4-stage pipelined Normalized filter by Constrained design

The feedback loop multiply-add operations can be pipelined by redistributing the delays. Roundoff errors of these pipelined normalized lattice filters and those obtained by using look-ahead method to the nonpipelined Chebyshev filter transfer function are calculated using the method introduced in [4], and are compared in Table 1. Roundoff noise gain factors of the direct-form cascade filters are also given. As can be seen from the table, better roundoff error characteristics are obtained by using constrained filter design. Roundoff noise gain factors of basic and minimum-noise lattice filters are given in Table 2. Note that the roundoff errors of minimum-noise lattice filters decrease as the pipelining stage increases.

Table 1 Roundoff noise gains of Normalized lattice and Direct-form cascade filters

	Normalized lattice		Cascade
	Constrained	Look-ahead	Look-ahead
nonpipe.	5.5246	5.8544	19874.1
2-stage	8.9003	8.1268	20211.9
4-stage	8.3172	16.0518	20227.5
8-stage	16.0489	32.0286	20340.6

Table 2 Roundoff noise gains of Normalized and Minimum-noise lattice filters

	Basic lattice	Minimum-noise
	Constrained	Constrained
nonpipe.	4.5225	2.1524
2-stage	9.0016	2.0216
4-stage	8.5022	1.1348
8-stage	16.1041	0.2815

The number of multipliers and adders needed for the implementation of these filters by normalized lattice filters are compared in Table 3. Note that the 4th order filter by constrained design could be implemented with one complex pole pair, and requires fewer multipliers than all other filters. The table shows that the lattice filters obtained by using the constrained filter design method require less hardware complexity than those obtained using look-ahead on the nonpipelined filter. Table 4 shows the number of multipliers and adders needed for the implementation of basic lattice and minimum-noise lattice filters obtained by using the constrained filter design method.

Table 3 Number of Multipliers and adders of Normalized lattice and Direct-form cascade filters

	Normalized latice				Cascade	
	Constrained		Look-ahead		Look-ahead	
	Multi	Adder	Multi	Adder	Multi	Adder
nonpipe.	19	11	19	11	5	8
2-stage	23	15	23	15	9	12
4-stage	15	11	31	23	13	16
8-stage	23	19	47	39	17	20

Table 4 Number of Multipliers and Adders of Basic and Minimum-noise lattice filters

	Basic lattice (Constrained)		Minimum-noise (Constrained)	
	Multi.	Adder	Multi.	Adder
nonpipe.	13	12	22	11
2-stage	17	16	30	15
4-stage	13	12	22	11
8-stage	21	20	38	19

6. Conclusions

In this paper, it was shown that the lattice filter structures in [1], [2] and [4] are suitable for pipelining. By using constrained filter design and the lattice filter design procedures, we designed pipelined high-speed lattice filters with extremely small roundoff errors. The hardware complexity of these structures is also much less than the pipelined lattice filters obtained by using scattered look-ahead on the nonpipelined filters.

As for further research work, the development of new pipelined lattice filter structures is also of interest. Wave digital filters also possess very good finite word length properties. Design of pipelined wave digital filters needs to be addressed in future. In [13] - [14], most significant bit first redundant arithmetic has been used to pipeline recursive and wave digital filters. Our pipelined filters can be used in combination with the approaches presented in [13] - [14] also.

References

[1] A. H. Gray, Jr. and J. D. Markel, "Digital Lattice and Ladder Filter Synthesis", *IEEE Trans. on Audio and Electroacoustics*, pp. 491-500, Dec. 1973

[2] A. H. Gray, Jr. and J. D. Markel, "A Normalized Digital Filter", *IEEE Trans. on Acoustics, speech, and signal processing*, vol. ASSP-33, pp. 258-277, June 1975

[3] S. K. Rao and T. Kailath, "Orthogonal Digital Filters for VLSI Implementation", *IEEE Trans. on Circuits and systems*, pp. 933-945, Nov. 1984

[4] S. M. Lei and K. Yao, "A Class of Systolizable IIR Digital Filters and Its Design for Proper Scaling and Minimum Output Roundoff Noise", *IEEE Trans. on Circuits and systems*, pp. 1217-1230, Oct. 1990

[5] S. Y. Kung, "On supercomputing with systolic/ wavefront array processors", *Proc. IEEE*, vol. 72, pp. 867-884, July 1984

[6] K. K. Parhi and D.G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters-Part I: Pipelining Using Scattered Look-Ahead and Decomposition", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, pp. 1099-1117, July 1989

[7] A. G. Deczky,"Synthesis of recursive digital filters using the minimum p-error criterion", *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 257-263, Oct. 1972

[8] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization", *Comput. J.*, vol. 6, pp. 163-168, 1963

[9] M. A. Richards, "Application of Deczky's program for recursive filter design to the design of recursive decimators", *IEEE Trans. on ASSP*, Oct. 1982

[10] A. G. Deczky, "Program for minimum-p synthesis of recursive digital filters" in *Programs for digital signal processing*, New York: IEEE Press, sec. 6.2 , 1979

[11] J. Schur, "Ueber Potenzreihen die im Innern des Einheitslreises beschrankt sind", *J. Reine Angewandte Mathematik*, vol. 147, pp. 205-232, Berlin 1917

[12] A. V. Oppenheim and R.W. Schafer, *Discrete Time Signal Processing*, Prentice-Hall, 1989

[13] S. C. Knowles, J. G. McWhirther, R. F. Woods, and J. V. McCanny, "Bit-level Systolic Arithmetics for High Performance IIR Filtering", *Journal of VLSI Signal Processing*, Kluwer Academic Press, vol. 1, 1989

[14] R. J. Singh and J. V. McCanny, "A Wave Digital Filter Three-Port Adaptor with Fine-Grained Pipelining", *Proc. of the 1991 Applications-Specific Array Processors*, Sept. 1991, Costa Brava (Spain), IEEE Computer Society Press, pp. 116-128.